CLAIMS

We claim:

[c1] 1. A method in a computer system for accessing a collection of data items, the method comprising:

when adding a data item to the collection,

fetching and adding to a write counter, the fetched write counter pointing to a bucket within a bucket array;

reading from the bucket pointed to by the fetched write pointer using a synchronization access mode of sync;

storing the data item in association with the bucket pointed to by the fetched write pointer;

writing to the bucket pointed to by the fetched write pointer using a synchronization access mode of sync; and

fetching and adding to a lower bound to indicate the number of data items added to the collection.

[c2] 2. The method of claim 1 wherein the bucket pointed to by the fetched write pointer contains a pointer to a linked list of data items.

[c3] 3. The method of claim 1 wherein the fetched write pointer modulo a number of buckets in the bucket array points to a bucket within the bucket array.

[c4] 4. The method of claim 1 wherein the adding adds one to the write counter.

[c5] 5. The method of claim 1 wherein the adding adds a size of a bucket to the write counter.

[c6] 6. The method of claim 1 including

when removing a data item from the collection,

fetching and adding to a read counter, the fetched read counter pointing to a bucket within the bucket array;

reading from the bucket pointed to by the fetched read pointer using a synchronization access mode of sync;

removing the data item from association with the bucket pointed to by the fetched read pointer; and

writing to the bucket pointed to by the fetched write pointer using a synchronization access mode of sync.

[c7] 7. The method of claim 1 including

when removing a data item from the collection,

checking the lower bound to ensure that the collection contains a data item,

when it cannot be ensured that the collection contains a data item, indicating that the collection may be empty; and

when it can be ensured that the collection contains a data item,

fetching and adding to a read counter, the fetched read counter pointing to a bucket within the bucket array;

reading from the bucket pointed to by the fetched read pointer using a synchronization access mode of sync;

removing the data item from association with the bucket pointed to by the fetched read pointer; and

writing to the bucket pointed to by the fetched write pointer using a synchronization access mode of sync.

[c8] 8. The method of claim 7 wherein the checking includes fetching and adding a negative number to the lower bound.

[c9] 9. The method of claim 8 wherein the checking includes fetching and adding a positive number to the lower bound when it cannot be ensured that the collection contains an item.

[c10] 10. The method of claim 1 wherein the synchronization access mode of sync prevents simultaneous access of a bucket by multiple threads.

[c11] 11. The method of claim 1 wherein the collection of data items is stored in an array data structure.

[c12] 12. The method of claim 1 wherein the collection of data items is stored in a linked list data structure.

[c13] 13. The method of claim 1 wherein the collection of data items is stored in a tree data structure.

[c14] 14. The method of claim 1 wherein the reading is permitted only when the bucket is full.

[c15] 15. The method of claim 14 wherein after the reading, the bucket is set to empty.

[c16] 16. The method of claim 1 wherein the writing is permitted only when the bucket is empty.

[c17] 17. The method of claim 16 wherein after the writing, the bucket is set to full.

[c18] 18. A method in a computer system for accessing a collection of data items, the method comprising:

when removing a data item from the collection,

fetching and adding to a read counter, the fetched read counter pointing to a bucket within the bucket array;

reading from the bucket pointed to by the fetched read pointer using a synchronization access mode of sync;

removing the data item from association with the bucket pointed to by the fetched read pointer; and

writing to the bucket pointed to by the fetched write pointer using a synchronization access mode of sync.

[c19] 19. The method of claim 18 including before fetching and adding to the read counter checking a lower bound to ensure that the collection contains a data item.

[c20] 20. The method of claim 19 wherein it cannot be ensured that the collection contains a data item, indicating that a data item cannot be removed.

[c21] 21. The method of claim 18 wherein the bucket pointed to by the fetched read pointer contains a pointer to a linked list of data items.

[c22] 22. The method of claim 18 wherein the fetched read pointer modulo a number of buckets in the bucket array points to a bucket within the bucket array.

[c23] 23. The method of claim 18 wherein the removing adds one to the read counter.

[c24] 24. The method of claim 18 wherein the adding adds a size of a bucket to the read counter.

[c25] 25. The method of claim 18 wherein the synchronization access mode of sync prevents simultaneous access of a bucket by multiple threads.

[c26] 26. The method of claim 18 wherein the collection of data items is stored in an array data structure.

[c27] 27. The method of claim 18 wherein the collection of data items is stored in a linked list data structure.

[c28] 28. The method of claim 18 wherein the collection of data items is stored in a tree data structure.

[c29] 29. The method of claim .18 wherein the reading is permitted only when the bucket is full.

[c30] 30. The method of claim 29 after the reading, the bucket is set to empty.

[c31] 31. The method of claim 18 wherein the writing is permitted only when the bucket is empty.

[c32] 32. The method of claim 31 wherein after the writing, the bucket is set to full.

[c33] 33. A method in a computer system for accessing a collection of data items, the method comprising:

defining a pointer to indicate a location for a data item;

defining a lower bound to indicate a number of items in the collection; and

determining based on the lower bound whether the collection has a data item; and

when it is determined that the collection has a data item,

adjusting the lower bound to indicate that the collection has one less data item; and accessing the data item at the location defined by the pointer using an access mode of sync.

[c34] 34. The method of claim 33 wherein read access to the location is permitted only when the location is full.

[c35] 35. The method of claim 34 after the read access, the location is set to empty.

[c36] 36. The method of claim 33 wherein write access to the location is permitted only when the location is empty.

[c37] 37. The method of claim 36 wherein after the write access, the location is set to full.

[c38] 38. The method of claim 33 wherein the data items of the collection are accessed by multiple readers and writers.

[c39] 39. The method of claim 33 wherein the data items of the collection are accessed by multiple producers.

[c40] 40. The method of claim 33 wherein the data items of the collection are accessed by multiple consumers.

[c41] 41. The method of claim 33 including

when access to the location by a thread is blocked,

enabling an exception to be raised when the location is next accessed; and

blocking the thread; and

when an exception is raised as a result of access by another thread to that location,

completing the access by that other thread to that location; and

restarting execution of the blocked thread.

[c42] 42. The method of claim 41 wherein when access by the thread to the location is blocked, saving a state of the thread and storing a reference to the thread in the location.

[c43] 43. The method of claim 42 wherein the reference is a pointer to a data structure that identifies the blocked thread and the saved state.

[c44] 44. The method of claim 43 wherein the data structure indicates the value that was stored in the location before storing the reference.

[c45] 45. The method of claim 33 wherein the collection includes an array of buckets, each bucket including a pointer to data items.

[c46] 46. The method of claim 45 wherein the collection includes a write pointer that indicates a bucket into which a next data item is to be stored and a read pointer that indicates a bucket from which a next data item is to be read.

[c47] 47. The method of claim 45 wherein the pointed to data items are stored in a linked list.

[c48] 48. The method of claim 45 wherein multiple readers and writers can be accessing data items of different buckets simultaneously.

[c49] 49. A computer system for accessing data, comprising:

a collection of buckets;

a read counter and a write counter that point to buckets within the collection;

a data structure for each bucket with locations for holding data at each of the buckets; and

a lower bound indicating whether a data item is currently stored in the data structure.

[c50] 50. The computer system of claim 45 wherein reading data from a bucket is permitted only when the bucket is full.

[c51] 51. The computer system of claim 45 wherein write access to a bucket is permitted only when the bucket is empty.

[c52] 52. The computer system of claim 45 including accessing programs that each operates in a different thread.

[c53] 53. The computer system of claim 45 wherein the data is accessed by multiple reading threads and writing threads.

[c54] 54. The computer system of claim 45 including

when access by a thread to a bucket is blocked,

enabling an exception to be raised when the bucket is next accessed; and

blocking the thread; and

when an exception is raised as a result of access by another thread to that bucket,

completing the access by that other thread to that bucket; and

restarting execution of the blocked thread.

[c55] 55. A computer system for accessing a collection of data items, comprising:

fetching a lower bound by a reader of a data item, the lower bound indicating a number of data items; and

when the fetched lower bound is greater than or equal to one,

refetching and decrementing the lower bound;

when the refetched lower bound is less than one, incrementing the lower bound; and

when the refetched lower bound is not less than one, selecting one of a number of buckets, each bucket for referencing some of the data items.

[c56] 56. The method of claim 55 wherein the selecting includes calculating a bucket location based on a read counter modulo a number of buckets.

[c57] 57. The method of claim 55 wherein the decrementing subtracts one.

[c58] 58. The method of claim 55 wherein the incrementing adds one.

[c59] 59. The computer system of claim 55 wherein multiple producers and multiple consumers are accessing the collection of data items.